

Working paper

A generalized formulation for vehicle routing problems

Pedro Munari

Production Engineering Department, Federal University of São Carlos,

Rodovia Washington Luís - Km 235, CEP: 13565-905, São Carlos-SP, Brazil

munari@dep.ufscar.br

June, 2016

Abstract

Different types of formulations are proposed in the literature to model vehicle routing problems. Currently, the most used ones can be fitted into two classes, namely vehicle flow formulations and set partitioning formulations. These types of formulations differ from each other not only due to their variables and constraints but also due to their main features. Vehicle flow formulations have the advantage of being compact models, so general-purpose optimization packages can be used to straightforwardly solve them. However, they typically show weak linear relaxations and have a large number of constraints. Branch-and-cut methods based on specialized valid inequalities can also be devised to solve these formulations, but they have not shown to be effective for large-scale instances. On the other hand, set partitioning formulations have stronger linear relaxations, but requires the implementation of sophisticated techniques such as column generation and specialized branch-and-price methods. Due to all these reasons, so far it has been recognized in the vehicle routing community that these two types of formulations are rather different. In this paper, we show that they are actually strongly related as they correspond to special cases of a generalized formulation of vehicle routing problems.

1 Introduction

The literature on vehicle routing problems has become very rich and covers nowadays a variety of applications, modeling approaches and solution methods [30]. Due to their huge importance in practice, these problems have called the attention of many researchers and motivated a large number of collaborations between companies and academia [13]. In addition, vehicle routing problems lead to challenging formulations that require the development of sophisticated solution strategies and motivates the design of clever heuristics and meta-heuristics [2, 18].

Vehicle routing problems are typically modeled using two different types of formulations. The first type, known as vehicle flow (VF) formulation, is based on binary variables associated to arcs of a network representation of the problem. In general, this is more intuitive and leads to a compact model that can be straightforwardly put on a black-box optimization solver. Also, valid inequalities and constraints (most of them exponential in terms of the number

of customers) have been used to achieve a more effective strategy, resulting in specialized branch-and-cut methods. However, even with the use of very elaborate inequalities, VF formulations may be still very challenging for current optimization solvers. The main reason is the weak linear relaxation of these formulations.

A stronger linear relaxation is observed in the second type of models, known as set partitioning (SP) formulation. The number of constraints in this formulation is much smaller with respect to a VF formulation, but it has a huge number of variables: one for each feasible route in the problem. In the vast majority of cases, generating all these routes is not viable and hence the column generation technique is required to generate columns in an iterative way. Columns correspond to an incidence vector of feasible routes, which are generated by solving a resource constrained shortest path problem (RCESPP). Most implementations solve the RCESPP by a label-setting algorithm, which is aided with clever strategies to improve its performance [20, 6]. The solution strategies based on SP formulations are currently the most efficient to obtain optimal solutions of vehicle routing problems [2, 23]. Still, the performance can be very poor on problems that allow long routes, i.e. routes that visit many customers.

From this brief description of the two most used types of VRP formulations, we can observe that they have many opposite features and then can be recognized as very different from each other. However, in this paper we show that they are not so different, as they are actually special cases of a general formulation of vehicle routing problems, which we call as p -step formulation. In fact, this is a family of formulations, as different values of p lead to different formulations. We show that the VF formulation and the SP formulation are p -step formulations with particular choices of p . In addition, we prove a relationship between the bounds provided by the linear relaxation of p -step formulations with different p . Column generation can also be used to solve a p -step formulation, with the advantage that more dual information is sent to the RCESPP with respect to SP formulations.

The p -step formulation associate variables to partial paths in the network representation of the problem. This has the potential of reducing the difficulty of solving problems that allow long routes, the big challenge in a SP formulation. On the other hand, p -step formulations may lead to stronger linear relaxations than a VF formulation, the main weakness of this latter. Many other advantages can be achieved by using a p -step formulation, as we enunciate ahead in this paper.

A formulation based on partial paths has also been proposed in [24], for the VRP with time windows. Similarly to the p -step formulation, the partial paths can start and end at any node of the network and must visit exactly a given number of customers. The authors obtain this formulation by applying Dantzig-Wolfe decomposition to a modified vehicle flow formulation of the problem, which relies on a modified graph to represent the solution as a giant tour. They prove that the linear relaxation of the resulting model provides a bound that is larger than or equal to the bound provided by the standard two-index flow formulation. The relationship between formulations with different resource bounds is not analyzed by the authors and no computational experiments are reported for the proposed formulation.

A similar idea has also been applied to other types of problem. In [11], the authors propose a formulation based on horizon decomposition for the capacitated lot sizing problem

with setup times. They partition the time horizon in several subsets, possibly with overlap, to have smaller subproblems, so they can be quickly solved by a black-box optimization solver. In the column generation framework, columns become associated to production plans defined for only one of the partitions of the time horizon. These partial production plans are then combined in the master problems, as in the p -step formulation.

The remainder of this paper has the following structure. In Section 2, we review the vehicle flow and set partitioning formulations and quickly discuss about their main features. In Section 3, we propose the p -step formulation and present theoretical and computational results that relate the formulations obtained using different choices of p . The column generation scheme for p -step formulations is proposed in Section 4, followed by a discussion of its main advantages and disadvantages in Section 5.

2 Classical VRP formulations

In this section, we review the formulations of two classical VRP variants, namely the capacitated vehicle routing problem (CVRP) and the vehicle routing problem with time windows (VRPTW). These formulations are the basis for many other variants of the problem. The purpose is to set the notation, nomenclature and foundations for the remaining sections of this paper.

Consider a set of customers represented by $\mathcal{C} = \{1, \dots, n\}$, such that a positive demand is associated to each customer $i \in \mathcal{C}$. To service these customers, we have to design routes for a fleet with K vehicles available in a single depot. Each route must start at the depot, visit a subset of customers and then return to the depot. All customers must be visited exactly once. Each vehicle has a maximum capacity Q , which limits the number of customers it can visit before returning to the depot. For the sake of clarity, we assume a homogeneous fleet of vehicles, but the discussion presented ahead can be easily extended to a heterogeneous fleet.

We represent the problem using a graph $G(\mathcal{N}, \mathcal{E})$, in which $\mathcal{N} = \mathcal{C} \cup \{0, n+1\}$ is the set of nodes associated to customers in \mathcal{C} and to the depot nodes 0 and $n+1$. We use two nodes to represent the same single depot and impose that all routes must start on 0 and return to $n+1$. Set \mathcal{E} contains the arcs (i, j) for each pair of nodes $i, j \in \mathcal{N}$ (we assume a complete graph). The cost of crossing an arc $(i, j) \in \mathcal{E}$ is denoted by c_{ij} . Each node has a demand q_i , such that $q_i > 0$ for each $i \in \mathcal{C}$ and $q_0 = q_{n+1} = 0$. The objective of the problem is to determine a set of minimal cost routes that satisfies all the requirements defined above.

2.1 Two-index vehicle flow formulation

In the two-index VF formulation, we define the binary decision variable x_{ij} that assumes value 1 if and only if there is a route that goes from customer i to j directly, for $i, j \in \mathcal{N}$. In addition, y_j is a continuous decision variable corresponding to the cumulated demand on the route that visits node $j \in \mathcal{N}$ up to this visit. With these parameters and decision variables,

the two-index flow formulation of the CVRP is given by:

$$\min \quad \sum_{i=0}^{n+1} \sum_{j=0}^{n+1} c_{ij} x_{ij} \quad (2.1)$$

$$\text{s.t.} \quad \sum_{\substack{j=1 \\ j \neq i}}^{n+1} x_{ij} = 1, \quad i = 1, \dots, n, \quad (2.2)$$

$$\sum_{\substack{i=0 \\ i \neq h}}^n x_{ih} - \sum_{\substack{j=1 \\ j \neq h}}^{n+1} x_{hj} = 0, \quad h = 1, \dots, n, \quad (2.3)$$

$$\sum_{j=1}^n x_{0j} \leq K, \quad (2.4)$$

$$y_j \geq y_i + q_j x_{ij} - Q(1 - x_{ij}), \quad i, j = 0, \dots, n+1, \quad (2.5)$$

$$d_i \leq y_i \leq Q, \quad i = 0, \dots, n+1, \quad (2.6)$$

$$x_{ij} \in \{0, 1\}, \quad i, j = 0, \dots, n+1. \quad (2.7)$$

Constraints (2.2) ensure that all customers are visited exactly once. Constraints (2.3) guarantee the correct flow of vehicles through the arcs, by stating that if a vehicle arrives to a node $h \in \mathcal{N}$, then it must depart from this node. Constraint (2.4) limits the maximum number of routes to K , the number of vehicles. Constraints (2.5) and (2.6) ensure together that the vehicle capacity is not exceeded. The objective function is defined by (2.1) and imposes that the total travel cost of the routes is minimized.

Constraints (2.5) also avoid subtours in the solution, *i.e.* cycling routes that do not pass through the depot. Different types of constraints are proposed in the literature to impose vehicle capacities and/or avoid subtours [17]. The advantage of using (2.5) and (2.6) is that the model has a polynomial number of constraints in terms of the number of customers. However, the lower bound provided by the linear relaxation of this model is known to be weak in relation to other models. Hence, many authors recur to capacity constraints that results in better lower bounds, even though the number of constraints becomes exponential in terms of the number of customers, requiring the use of a branch-and-cut strategy [28].

The VRPTW is an extension of the CVRP, in which customer time windows are imposed for the visits. A time window corresponds to a time interval $[w_i^a, w_i^b]$ which imposes that the service at node $i \in \mathcal{N}$ cannot start earlier than the time instant w_i^a nor later than w_i^b . If the vehicle arrives before than w_i^a , then it has to wait until this instant to start servicing the node. To each arc $(i, j) \in \mathcal{E}$, we assign a travel time t_{ij} , which respects triangle inequality. Also, each node i has a service time t_i that corresponds to the minimum amount of time that the vehicle has to stay in a visited node.

Let w_i be a continuous decision variable representing the time instant that the service starts at node $i \in \mathcal{N}$. We obtain a model for the VRPTW by adding the following constraints to the formulation (2.1)–(2.7):

$$w_j \geq w_i + (s_i + t_{ij})x_{ij} - M_{ij}(1 - x_{ij}), \quad i = 0, \dots, n; \quad j = 1, \dots, n+1, \quad (2.8)$$

$$w_i^a \leq w_i \leq w_i^b, \quad i = 0, \dots, n+1, \quad (2.9)$$

where M_{ij} is a sufficiently large value, which can be defined as $M_{ij} = \max\{w_i^b - w_j^a, 0\}$.

2.2 Set partitioning formulation

Currently, the most efficient exact methods for solving VRP variants are based on SP formulations. The variables in these formulations correspond to feasible routes of the problem. Let \mathcal{R} be the set of routes that satisfy the problem requirements. For example, in the CVRP, a route in \mathcal{R} must start and finish at the depot, visit at most once a customer, respect the vehicle capacity and guarantee that if the route arrives to a customer then it has to leave this customer. The same requirements are valid for the VRPTW, in addition to satisfying time windows of all visited customers.

Let λ_r be the binary decision variable that is equal to 1 if and only if the route $r \in \mathcal{R}$ is selected. The SP formulation is as follows:

$$\min \quad \sum_{r \in \mathcal{R}} c_r \lambda_r \quad (2.10)$$

$$\text{s.t.} \quad \sum_{r \in \mathcal{R}} a_{ri} \lambda_r = 1, \quad i \in \mathcal{C}, \quad (2.11)$$

$$\sum_{r \in \mathcal{R}} \lambda_r \leq K, \quad (2.12)$$

$$\lambda_r \in \{0, 1\}, \quad r \in \mathcal{R}. \quad (2.13)$$

This can be used to model the CVRP, VRPTW and many other VRP variants, depending on how we define the set of routes \mathcal{R} . The objective function (2.10) minimizes the total cost of the selected routes. The cost of route $r \in \mathcal{R}$, denoted by c_r , is computed using the arc costs c_{ij} defined above. Namely, given a route r that sequentially visits nodes i_0, i_1, \dots, i_p , $p > 0$, its total cost is given by

$$c_r = \sum_{j=0}^{p-1} c_{i_j i_{j+1}}, \quad (2.14)$$

Constraints (2.11) impose exactly one visit to each customer node. Each column $a_r = (a_{r1}, \dots, a_{rn})^T$ is a binary vector in which $a_{ri} = 1$ if and only if the corresponding route r visits customer i . Constraint (2.12) imposes the maximum number of vehicles available at the depot. If K is sufficiently large for the problem, then this constraint can be dropped from the formulation.

Generating all routes of \mathcal{R} is impractical in general, as the number of routes is exponential in terms of the numbers of customers. Hence, set partitioning formulations require using the column generation technique for solving the linear relaxation of model (2.10)–(2.13) [19]. As a consequence, to obtain optimal integer solutions we need a branch-and-price method [25]. In the column generation technique, we start with a small subset of routes $\bar{\mathcal{R}} \subset \mathcal{R}$ that is

used to create the following restricted master problem (RMP):

$$\min \sum_{r \in \overline{\mathcal{R}}} c_r \lambda_r \quad (2.15)$$

$$\text{s.t. } \sum_{r \in \overline{\mathcal{R}}} a_{ri} \lambda_r = 1, \quad i \in \mathcal{C}, \quad (2.16)$$

$$\sum_{r \in \overline{\mathcal{R}}} \lambda_r \leq K, \quad (2.17)$$

$$\lambda_r \geq 0, \quad r \in \overline{\mathcal{R}}. \quad (2.18)$$

Notice that the RMP is the linear relaxation of (2.10)–(2.13), but considering only a subset of variables. Let $u = (u_1, \dots, u_n) \in \mathbb{R}^n$ and $\sigma \in \mathbb{R}_-$ be the dual variables associated to constraints (2.16) and (2.17), respectively. At each iteration of the column generation method, we solve the RMP to obtain a dual solution $(\bar{u}, \bar{\sigma})$ that is used to generate the columns that are not in the RMP yet. These columns are associated with feasible routes obtained by solving the following subproblem:

$$\min_{r \in \mathcal{R}} rc(\bar{u}, \bar{\sigma}) = \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} (c_{ij} - \bar{u}_i) x_{rij} - \bar{\sigma} \quad (2.19)$$

where $\bar{u}_0 = \bar{u}_{n+1} = 0$ and $x_r = \{x_{rij}\}_{i,j \in \mathcal{N}}$ is a binary vector such that $x_{rij} = 1$ if and only if route $r \in \mathcal{R}$ visits node i and goes directly to node j . This subproblem is a *resource constrained elementary shortest path problems* (RCESPP) [16]. Let \bar{x}_r be associated to an optimal route r of the subproblem. If the corresponding value $rc(\bar{u}, \bar{\sigma})$ is negative, then a new variable λ_r can be added to the RMP using this route. Indeed, $rc(\bar{u}, \bar{\sigma})$ is the reduced cost of this new variable, for which we have the following cost and column coefficients:

$$c_r := \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} c_{ij} \bar{x}_{rij},$$

$$a_{ri} := \sum_{j \in \mathcal{N}} \bar{x}_{rij}, \quad i \in \mathcal{C}.$$

Hence, r is added to $\overline{\mathcal{R}}$ and the new RMP has to be solved again. If $rc(\bar{u}, \bar{\sigma})$ is nonnegative and $(\bar{u}, \bar{\sigma})$ are optimal dual solutions of the current RMP, then the optimal solution of the current RMP is also optimal for the linear relaxation of the MP. Hence, the column generation method terminates successfully.

The performance of a computational implementation of the column generation algorithm is strongly dependent on the way RMPs and subproblems are solved. To be successful, implementations should quickly solve the RMPs and use stable dual solutions that help to reduce the total number of iterations [19, 21, 22]. Solving the RCESPP effectively is also a very important requirement in a column generation algorithm for VRP variants. Although integer programming formulations are available for the RCESPP, they cannot be solved effectively by the current state-of-the-art optimization solvers [26]. The current best strategies use a label-setting algorithm based on dynamic programming. This algorithm was originally proposed

by [9] and [3] and since then has been continuously improved [10, 27, 5, 7, 1, 20, 6].

3 A family of vehicle routing problem formulations

In this section, we propose a generalized VRP formulation. The idea of this new family of formulations is to have binary variables associated to up to p (sequential) steps in the network. One step corresponds to traversing a given arc in the network, so p steps correspond to a partial path that traverses exactly p arcs. Let \mathcal{S} be the set of all feasible p -steps in the network, including also all the feasible k -step paths that start at node 0, for all $k = 1, \dots, p-1$, when $p > 1$. By feasible we mean that the arcs can be traversed sequentially and that none of the resources are violated. For instance, if $p = 2$ then \mathcal{S} is the set of all partial paths of the forms $i - j - k$ and $0 - j$, for any $i, j, k \in \mathcal{N}$. Given a partial k -step path $s \in \mathcal{S}$, for $k = 1, \dots, p$, we denote by i_s and j_s its first and last nodes, respectively. Let λ_s be a binary variable that is equal to 1 if and only if the arcs in $s \in \mathcal{S}$ are traversed sequentially in the optimal path. For the sake of clarity, we assume at first that capacity is the only resource in the problem. Let φ_j be a continuous decision variable that is equal to the cumulated demand of all nodes visited by a route up to node j (inclusive). The p -step formulation for the CVRP is as follows:

$$\min \quad \sum_{s \in \mathcal{S}} c_s \lambda_s \quad (3.1)$$

$$\text{s.t.} \quad \sum_{s \in \mathcal{S}} e_i^s \lambda_s = 1, \quad i = 1, \dots, n, \quad (3.2)$$

$$\sum_{s \in \mathcal{S}} a_i^s \lambda_s = 0, \quad i = 1, \dots, n, \quad (3.3)$$

$$\sum_{s \in \mathcal{S}} a_0^s \lambda_s \leq K, \quad (3.4)$$

$$\varphi_j \geq \varphi_i + q_j \sum_{s \in \mathcal{S}_{ij}} \lambda_s - Q(1 - \sum_{s \in \mathcal{S}_{ij}} \lambda_s), \quad i = 0, \dots, n, \quad j = 1, \dots, n+1, \quad (3.5)$$

$$q_i \leq \varphi_i \leq Q, \quad i = 1, \dots, n, \quad (3.6)$$

$$\lambda_s \in \{0, 1\}, \quad s \in \mathcal{S}, \quad (3.7)$$

where $\mathcal{S}_{ij} \subset \mathcal{S}$ contains only the paths that traverse arc (i, j) for a given pair $i, j \in \mathcal{N}$; c_s is the total cost of traversing all arcs in path s ; and a^s and e^s are vectors defined as

$$a_i^s = \begin{cases} +1, & \text{if } i \text{ is the first node visited by path } s, \\ -1, & \text{if } i \text{ is the last node visited by path } s, \\ 0, & \text{otherwise,} \end{cases} \quad (3.8)$$

$$e_i^s = \begin{cases} +1, & \text{if } i \text{ is visited by path } s, \text{ but it is not the last node of } s, \\ 0, & \text{otherwise,} \end{cases} \quad (3.9)$$

for all $i = 0, 1, \dots, n$ and $s \in \mathcal{S}$. In this formulation, constraints (3.2) impose that each customer node is visited at most once; constraints (3.3) ensure that if two paths are linked, then the last node in one path is the same as the first node in the other; constraint (3.4) imposes the maximum number of (complete) routes in an optimal solution; constraints (3.5)

and (3.6) ensure that routes satisfy the capacity resource and has no subtours; and (3.7) impose the binary domain of the decision variables λ . Notice that the p -step paths in \mathcal{S} must traverse exactly p arcs in the network. The only partial paths in \mathcal{S} that are allowed to cross less than p arcs are those that starts on the depot node 0.

Customer time windows can also be included in the p -step formulation by adding the following constraints, resulting in the VRPTW p -step model:

$$\omega_j \geq \omega_i + (s_i + t_{ij}) \sum_{s \in \mathcal{S}_{ij}} \lambda_s - M(1 - \sum_{s \in \mathcal{S}_{ij}} \lambda_s), \quad i = 0, \dots, n, \quad j = 1, \dots, n+1, \quad (3.10)$$

$$w_i^a \leq \omega_i \leq w_i^b, \quad i = 0, \dots, n+1,$$

where ω_i is a decision variable that indicates the time instant that the service starts at node $i = 0, \dots, n+1$.

Constraints (3.5) can be written in a coupled way, based only on the first (i_s) and last (j_s) nodes of the partial path $s \in \mathcal{S}$, as follows:

$$\varphi_{j_s} \geq \varphi_{i_s} + q_s \lambda_s - Q(1 - \lambda_s), \quad s \in \mathcal{S}, \quad (3.11)$$

$$q_s \leq \varphi_{i_s} \leq Q, \quad s \in \mathcal{S}. \quad (3.12)$$

where q_s is the total demand of the nodes visited by this path, except for its first node.

3.1 Special cases of the p -step family

As mentioned before, model (3.1)–(3.7) is a family of formulations, because for each $p = 1, \dots, n$ we have different paths. Indeed, for particular choices of p , we obtain the VF formulation (2.1)–(2.7) and the SP formulation (2.10)–(2.13), as presented in Proposition 3.1.

Proposition 3.1: The vehicle flow formulation (2.1)–(2.7) and the set partitioning formulation (2.10)–(2.13) are particular cases of the p -step formulation, with $p = 1$ and $p = n+1$, respectively.

Proof. In the 1-step formulation, the set \mathcal{S} is given by all the single arcs (i, j) , with $i, j = 0, 1, \dots, n+1$. Hence, all variables in this formulation can be rewritten as $\lambda^s = x_{i_s j_s}$, where (i_s, j_s) is the arc traversed by the 1-step path $s \in \mathcal{S}$. By replacing this in the p -step formulation (3.1)–(3.7) and noticing that path s can be expressed uniquely by its corresponding pair of nodes i_s and j_s , we obtain the VF formulation (2.1)–(2.7). On the other hand, in the $(n+1)$ -step formulation, set \mathcal{S} is given by all the k -paths that start at node 0, $1 < k < n+1$, as no $(n+1)$ -step path exists. Hence, \mathcal{S} can be reduced to all feasible complete routes, as in the usual SP formulation. In addition to that, since any path in $s \in \mathcal{S}$ is a proper feasible route, resource constraints in the p -step formulation are redundant and can be discarded, which leads to the SP formulation (2.10)–(2.13). \square

3.2 Intermediate p -step formulations

For any value of $p = 1, \dots, n+1$, the corresponding p -step formulation is a valid vehicle routing problem formulation. The basic difference between formulations with different values of p lies in the level of arc coupling in the partial paths. Indeed, in the 1-step formulation, the arcs are totally detached, so the model has to decide what is the best way of connecting them, without violating other constraints such as elementarity and resource availability. The number of variables in the model is polynomial in terms of the number of nodes, so a general-purpose optimization package could be used to solve it. In addition, generating these paths is quick and straightforward. However, the VF formulation is well known for its poor performance, mainly due to a weak linear relaxation. On the other, in the $(n+1)$ -step formulation all the arcs are already attached so the model has only to choose what is the best set of routes. The SP formulation is well recognized by having a stronger linear relaxation, but column generation and branch-and-price methods are required to solve the problem, as the number of variables is exponential in terms of the number of customers. In this case, the difficulty lies in generating the paths, as they must be feasible routes that depart from and return to the depot. These features illustrate that VF formulations and SP formulations are extremal cases of p -step formulations.

At this point, an intriguing question emerges: Is there a choice of p such that the p -step formulation has a reasonably strong linear relaxation and performs well in practice? Proposition 3.2 brings an interesting relationship between p -step formulation regarding their optimal values.

Proposition 3.2: Let \tilde{z}_p be the optimal value of the linear relaxation of a p -step formulation, for $p = 1, \dots, n+1$. For any $p \in \{1, \dots, n\}$, we have that $\tilde{z}_{p+1} \geq \tilde{z}_p$.

Proof. By contradiction, let us assume that $\tilde{z}_{p+1} < \tilde{z}_p$. The p -step formulation has the following partial paths:

$$\begin{aligned} &(0, v_1), \\ &(0, v_1, \dots, v_k), \text{ for } 1 < k \leq p, \\ &(v_0, v_1, \dots, v_p), \text{ with } v_0 > 0. \end{aligned}$$

In the $(p+1)$ -step formulation, we have the partial paths:

$$\begin{aligned} &(0, v_1), \\ &(0, v_1, \dots, v_k), \text{ for } 1 < k \leq p, \\ &(0, v_1, \dots, v_p, v_{p+1}), \\ &(v_0, v_1, \dots, v_p, v_{p+1}), \text{ with } v_0 > 0. \end{aligned}$$

Hence, both formulations have all k -step paths starting on 0, for $k = 1, \dots, p$. The difference between them lies in the partial paths starting on $v_0 > 0$ and in the $(p+1)$ paths starting on 0. Let λ^* and u^* be primal and dual optimal solutions of the p -step formulation. Since they are optimal, no partial path in the p -step formulation can lead to a column with negative

reduced cost for the dual solution u^* . On the other hand, since $\tilde{z}_{p+1} < \tilde{z}_p$, there must exist a path $(v_0, v_1, \dots, v_{p+1})$ in the $(p+1)$ -step formulation that leads to a column with negative reduced cost for u^* . However, this is a contradiction, as any p -step path visiting only vertices in $\{v_0, v_1, \dots, v_{p+1}\}$ has a reduced cost greater than or equal to zero. Indeed, the sum of $p+1$ terms cannot be negative if the sum of each combination having p terms of them is non-negative. Therefore, $\tilde{z}_{p+1} \geq \tilde{z}_p$. \square

We now present an empirical evidence of the difference in the optimal values for p -step formulations with different values of p . The experiments use the well known Solomon's instances of the VRPTW [29], considering only the first 25 customers. All partial paths were generated by full enumeration and we added only the feasible ones, i.e. partial paths that satisfy vehicle capacity and customer time windows requirements. The formulation with all generated paths was solved by the IBM CPLEX v.12.4 Optimization Studio, using the CPLEX Concert Library. The experiments were run on a Dell Precision T7600 with Intel E5-6280 2.70 GHz CPU and 192 GB RAM, under Linux Operating System.

Table 1 shows the optimal values regarding the linear relaxations of the p -step formulations with p from 1 to 10. As stated in Proposition 3.2, we observe that the optimal value of the linear relaxation for a given $p \in \{1, \dots, n\}$ is smaller than or equal to the optimal value for $p+1$. Tables 2 and 3 shows the corresponding total CPU time (in seconds) to solve the linear relaxation and the total number of columns in the problem, respectively.

The plots in Figures 1 to 3 summarize the results presented in the tables, for class R1. For the other classes the behavior is similar. In the plots on the left, the horizontal axis give the name of the instance, while the lines represent the behavior of a p -step formulation for a particular p . In the plots on the right, we have the opposite, so we can observe the behavior of an instance according to different values of p . The plots in Fig. 1 show the objective values of the linear relaxations. As expected, a line for a given value of p dominates all the other lines with a smaller value, for the plot on the left. An interesting behavior is observed in the plot on the right. The easiest instances (e.g. R101 and R102) are those with the largest difference on the values according to the variation of p . The most difficult instances (e.g. R112 and R108) are those with the smallest variations in objective values. The plots in Fig. 2 show that the largest computational times are for instances R104 and R108, with intermediate values of p (from 5 to 7). The same instances and values of p have the largest number of feasible partial paths, as presented in the plots in Fig. 2.

4 Column generation for the p -step formulations

Any p -step formulation can be seen as a column generation model, as any of its columns (variables) can be generated by following a known rule. Of course, for p small, the number of columns in the formulation is polynomial in terms of the number of nodes and hence it can be practical to enumerate them beforehand. Even so, for large-scale problems it can be more advantageous to recur to column generation, as only a few variables will be nonzero at the optimal solution.

Consider the linear relaxation of the p -step formulation (3.1)–(3.7) having only the columns

Instance	1	2	3	4	5	6	7	8	9	10
C101	179.59	184.02	186.28	187.93	188.20	188.20	188.46	189.26	190.56	191.30
C102	147.76	151.32	154.08	155.01	155.33	156.18	—	—	—	—
C103	125.25	129.95	134.03	134.24	134.24	—	—	—	—	—
C104	124.05	129.09	131.67	133.00	133.17	—	—	—	—	—
C105	76.85	88.94	107.33	116.74	129.36	138.47	152.84	159.83	164.17	169.26
C106	156.26	166.65	171.13	174.39	174.55	178.00	181.26	183.84	185.32	188.11
C107	74.82	85.34	92.50	99.05	108.94	119.43	125.35	136.94	145.16	153.02
C108	74.14	78.04	88.44	90.36	100.31	109.48	—	—	—	—
C109	73.05	76.03	84.56	89.23	97.29	103.78	—	—	—	—
R101	328.26	362.18	406.53	457.21	500.61	585.30	617.10	617.10	617.10	617.10
R102	306.99	312.13	331.24	366.54	407.88	450.61	494.93	520.76	531.68	546.33
R103	289.56	296.08	300.51	310.11	327.35	347.37	370.02	398.26	430.76	454.60
R104	282.49	291.66	293.90	299.31	305.41	321.19	335.35	352.98	377.71	406.76
R105	289.45	308.72	331.45	366.16	402.94	440.66	484.18	530.50	530.50	530.50
R106	286.67	297.49	305.66	326.10	339.77	357.35	387.70	421.72	447.15	457.30
R107	284.12	293.70	296.69	301.30	313.24	324.04	336.91	360.28	385.13	424.30
R108	282.01	290.78	292.88	296.01	299.89	303.96	312.36	330.71	350.02	376.21
R109	284.20	297.44	301.94	312.32	320.27	345.64	369.36	394.92	441.30	441.30
R110	282.57	292.01	297.52	303.09	310.49	328.31	349.95	371.77	410.04	438.35
R111	283.48	294.29	296.65	300.74	312.75	324.25	343.45	367.29	399.77	427.28
R112	281.49	290.69	290.71	291.60	293.76	298.80	302.43	318.79	334.29	371.75
RC101	95.58	105.46	118.82	134.60	173.55	210.58	251.17	300.86	366.26	406.63
RC102	94.67	101.61	111.77	120.37	127.52	143.56	167.99	219.93	295.42	311.35
RC103	94.63	99.45	105.20	109.13	111.68	117.66	139.96	199.58	267.76	293.82
RC104	94.61	99.43	104.24	106.18	108.09	110.38	123.47	184.08	242.67	270.43
RC105	98.67	103.44	112.49	121.81	145.81	176.10	211.54	270.03	346.43	377.02
RC106	94.21	101.89	107.72	113.49	129.80	151.56	183.42	229.84	290.69	323.68
RC107	94.05	97.85	103.38	106.14	110.33	112.49	133.40	182.91	238.11	270.88
RC108	93.86	97.61	100.69	103.57	104.50	105.05	109.42	173.70	229.50	267.45

Tab. 1: Optimal values of the linear relaxations of p -step formulations with $p = 1, \dots, 10$.

Instance	1	2	3	4	5	6	7	8	9	10
C101	0.01	0.08	0.32	3.07	9.77	18.71	27.93	22.03	17.44	14.69
C102	0.01	0.08	1.36	25.59	251.65	1576.00	—	—	—	—
C103	0.05	0.08	2.32	94.59	1009.28	—	—	—	—	—
C104	0.01	0.10	2.87	103.57	2189.55	—	—	—	—	—
C105	0.00	0.07	0.43	7.56	23.45	79.16	118.18	157.62	86.36	70.42
C106	0.01	0.06	0.38	3.44	13.85	24.82	37.07	40.26	21.32	18.38
C107	0.01	0.06	0.46	5.16	36.64	115.69	352.79	903.03	832.82	510.79
C108	0.01	0.08	0.75	11.27	97.29	568.18	—	—	—	—
C109	0.03	0.09	1.12	25.28	314.57	3544.28	—	—	—	—
R101	0.02	0.07	0.14	0.15	0.06	0.03	0.01	0.01	0.01	0.01
R102	0.01	0.09	1.03	11.26	30.86	28.18	11.89	3.34	1.44	0.84
R103	0.01	0.11	2.23	30.09	114.06	169.83	125.62	46.01	29.79	8.99
R104	0.01	0.10	3.67	56.44	347.91	879.46	867.75	280.54	181.03	138.56
R105	0.01	0.06	0.51	0.96	0.70	0.20	0.06	0.03	0.04	0.04
R106	0.01	0.09	1.50	23.49	67.94	98.66	38.47	16.16	6.20	2.88
R107	0.01	0.13	3.76	43.71	272.17	400.88	214.35	73.77	43.66	17.90
R108	0.01	0.10	5.43	67.78	578.89	1349.99	1074.32	477.24	231.88	205.78
R109	0.01	0.11	0.79	5.84	15.53	13.53	4.62	1.33	0.34	0.34
R110	0.00	0.12	2.27	18.88	70.33	55.21	31.23	8.57	3.92	2.02
R111	0.01	0.10	2.95	30.83	137.27	196.58	104.95	43.56	25.80	7.63
R112	0.00	0.13	2.87	43.47	190.15	459.44	253.17	84.80	58.26	27.44
RC101	0.01	0.05	0.20	0.42	0.69	0.56	0.31	0.13	0.08	0.04
RC102	0.01	0.09	0.50	3.18	10.05	21.94	30.28	25.64	12.79	3.43
RC103	0.01	0.10	0.85	8.26	39.29	83.67	197.94	106.78	70.90	28.01
RC104	0.01	0.12	1.33	18.30	104.77	320.58	631.76	1264.42	659.24	170.80
RC105	0.01	0.08	0.36	1.78	4.59	6.27	4.64	2.53	1.06	0.45
RC106	0.01	0.07	0.36	2.35	6.05	9.98	9.73	4.38	2.02	0.77
RC107	0.01	0.09	0.77	7.95	33.85	82.78	107.33	75.24	29.66	13.33
RC108	0.01	0.12	1.48	17.42	92.42	265.17	438.19	519.90	188.75	82.95

Tab. 2: Running times (in seconds) for solving the linear relaxations of p -step formulations with $p = 1, \dots, 10$.

Instance	1	2	3	4	5	6	7	8	9	10
C101	579	5803	34385	133467	354181	650788	824566	730563	527755	431334
C102	603	9339	109428	990829	6917245	36583123	141707550	381465687	665415185	702929285
C103	627	12084	198881	2756537	31907148	307392120	—	—	—	—
C104	627	13215	248679	4186918	60384382	765904269	—	—	—	—
C105	650	7498	52567	248296	825452	1955664	3256310	3700702	2970985	2187180
C106	580	5886	35696	142996	393496	750177	981876	889431	646196	526315
C107	651	7928	61563	340315	1410176	4469377	10726800	18496976	21291654	16768940
C108	651	9662	97318	721208	4055828	17547587	57493452	132263534	193418246	162430987
C109	651	11018	144385	1402004	10772852	65943878	316315086	1123552394	—	—
R101	623	4342	10317	10030	4512	1834	1560	1560	1560	1560
R102	634	8337	65739	292284	645797	663577	355597	193214	179176	179010
R103	648	11093	131574	941723	3199011	4606230	3144467	1589395	1339050	1332564
R104	648	12236	183132	1869440	9404019	19588934	18274415	10039471	7137406	6978228
R105	640	6217	27623	51824	41749	16441	8660	8328	8328	8328
R106	645	9801	93806	529603	1458479	1800303	1102398	554992	477315	475836
R107	651	11953	156784	1270626	4835114	7715253	5777954	2937523	2310954	2292146
R108	651	12963	204610	2226674	11763330	25593728	24972261	14126761	9756908	9473675
R109	651	8831	64337	246240	433028	348082	152708	92645	90200	90200
R110	651	11684	119395	597548	1348577	1381861	722546	366389	332508	332338
R111	651	11695	143843	1006377	3192454	4341908	2847092	1408300	1162757	1156982
R112	651	14935	232028	1740062	5714527	8208623	5716200	2844509	2273266	2256990
RC101	638	5698	22549	46644	56595	42858	22926	12951	10843	10748
RC102	650	8794	62019	248174	632617	1106851	1265406	896467	472783	337444
RC103	651	11485	118095	660844	2137758	4159425	4894645	3511669	1912280	1421480
RC104	651	12598	164665	1243642	5550180	15204179	24799695	22838754	13032058	8258749
RC105	646	7834	46786	155578	302611	350187	251154	134787	87680	80779
RC106	651	8233	54812	199703	423804	557133	453897	247418	144379	129965
RC107	651	11156	109140	608912	1985085	3983445	4615006	3004263	1481881	1162514
RC108	651	13729	173759	1212444	4858513	11922156	17352838	13841167	7164625	5074936

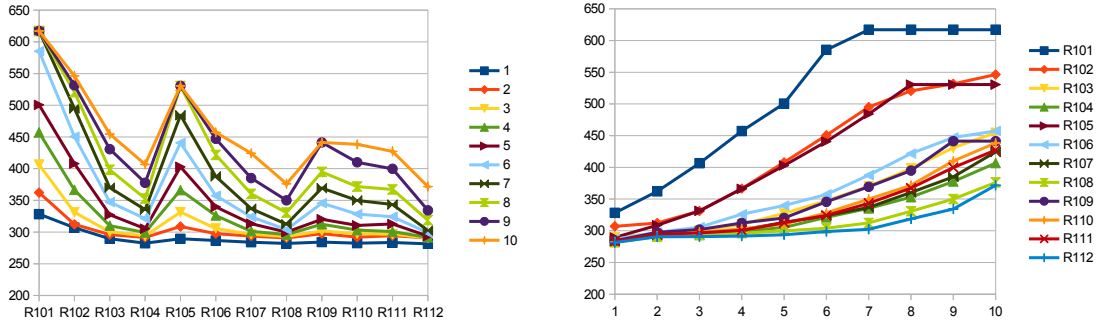
Tab. 3: Number of partial paths in the p -step formulations with $p = 1, \dots, 10$.

Fig. 1: Objective values of the linear relaxations of class R1.

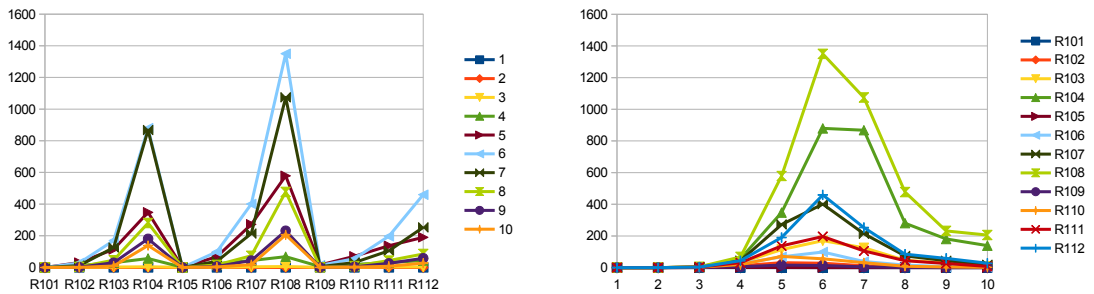


Fig. 2: Computational times to solve the linear relaxations for class R1.

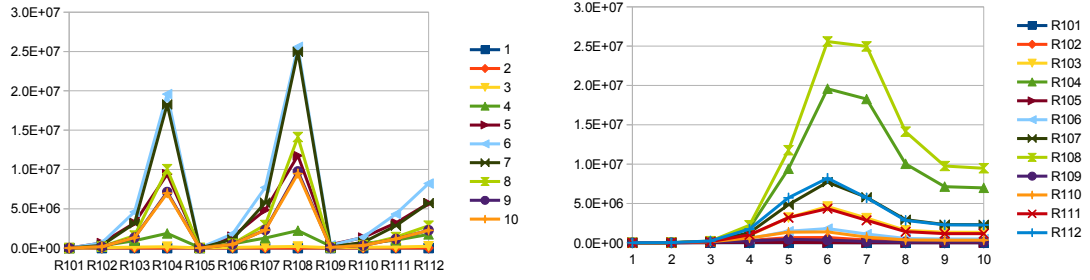


Fig. 3: Number of paths generated for class R1.

corresponding to an arbitrary subset $\bar{\mathcal{S}} \subset \mathcal{S}$. This leads to the following restricted master problem (RMP):

$$\min \quad \sum_{s \in \bar{\mathcal{S}}} c^s \lambda^s \quad (4.1)$$

$$\text{s.t.} \quad \sum_{s \in \bar{\mathcal{S}}} e_i^s \lambda^s = 1, \quad i = 1, \dots, n, \quad (4.2)$$

$$\sum_{s \in \bar{\mathcal{S}}} a_i^s \lambda^s = 0, \quad i = 1, \dots, n, \quad (4.3)$$

$$\sum_{s \in \bar{\mathcal{S}}} a_0^s \lambda^s \leq K, \quad (4.4)$$

$$\varphi_i - \varphi_j + \sum_{s \in \bar{\mathcal{S}}_{ij}} (q_j + Q) \lambda^s \leq Q, \quad i = 0, \dots, n, \quad j = 1, \dots, n+1, \quad (4.5)$$

$$q_i \leq \varphi_i \leq Q, \quad i = 1, \dots, n, \quad (4.6)$$

$$\lambda_s \geq 0, \quad s \in \bar{\mathcal{S}}, \quad (4.7)$$

where $\bar{\mathcal{S}}_{ij}$ has the same meaning as \mathcal{S}_{ij} , but considers only the paths in $\bar{\mathcal{S}}$. Notice that we have written constraints (4.5)–(4.7) in a slightly different way for the sake of clarity.

Let $u^1 = (u_1^1, \dots, u_n^1) \in \mathbb{R}^n$, $u^2 = (u_1^2, \dots, u_n^2) \in \mathbb{R}^n$, $u^3 \in \mathbb{R}$ and $u^4 = (u_{01}^4, u_{02}^4, \dots, u_{n,n+1}^4) \in \mathbb{R}^{n+1 \times n+1}$ be the dual variables associated to constraints (4.2)–(4.5), respectively. Given a dual solution $\bar{u} = (\bar{u}^1, \dots, \bar{u}^4)$ of the RMP, where we assume $\bar{u}_0^1 = \bar{u}_{n+1}^1 = 0$, the reduced cost of the column corresponding to a path (v_0, v_1, \dots, v_k) is given by:

$$\begin{aligned} rc(\bar{u}) &= \sum_{j=0}^{k-1} \left(c_{v_j v_{j+1}} - \bar{u}_{v_j}^1 - (q_{v_{j+1}} + Q) \bar{u}_{v_j v_{j+1}}^4 \right) \\ &\quad - \delta(v_0 \neq 0) \bar{u}_{v_0}^2 + \delta(v_k \neq n+1) \bar{u}_{v_k}^2 - \delta(v_0 = 0) \bar{u}^3, \end{aligned}$$

where $\delta(C)$ is equal to 1 if condition C is true; 0, otherwise.

Any feasible path with negative cost corresponds to a path in $\mathcal{S} \setminus \bar{\mathcal{S}}$. This can be used to generate a new column (variable) that has a negative reduced cost and then should be added to the current RMP. After solving the modified RMP a new dual solution is obtained and the

process is repeated. If it is not possible to find a path with negative cost, then the current optimal solution of the RMP is also optimal for the linear relaxation of (3.1)–(3.7).

Similarly, for the VRPTW, we have the following restricted master problem (RMP):

$$\min \quad \sum_{s \in \bar{\mathcal{S}}} c^s \lambda^s \quad (4.8)$$

$$\text{s.t.} \quad \sum_{s \in \bar{\mathcal{S}}} e_i^s \lambda^s = 1, \quad i = 1, \dots, n, \quad (4.9)$$

$$\sum_{s \in \bar{\mathcal{S}}} a_i^s \lambda^s = 0, \quad i = 1, \dots, n, \quad (4.10)$$

$$\sum_{s \in \bar{\mathcal{S}}} a_0^s \lambda^s \leq K, \quad (4.11)$$

$$\varphi_i - \varphi_j + \sum_{s \in \bar{\mathcal{S}}_{ij}} (q_j + Q) \lambda^s \leq Q, \quad i = 0, \dots, n, \quad j = 1, \dots, n+1, \quad (4.12)$$

$$q_i \leq \varphi_i \leq Q, \quad i = 1, \dots, n, \quad (4.13)$$

$$\omega_i - \omega_j + \sum_{s \in \bar{\mathcal{S}}_{ij}} (s_i + t_{ij} + M_{ij}) \lambda^s \leq M_{ij}, \quad i = 0, \dots, n, \quad j = 1, \dots, n+1, \quad (4.14)$$

$$w_i^a \leq \omega_i \leq w_i^b, \quad i = 0, \dots, n+1, \quad (4.15)$$

$$\lambda_s \geq 0, \quad s \in \bar{\mathcal{S}}, \quad (4.16)$$

where M_{ij} is a sufficiently large constant, e.g. $M_{ij} = w_i^b - w_j^a$. As in the CVRP formulation, we have dual variables u^1 to u^4 corresponding to constraints (4.10) to (4.12). Additionally, $u^5 = (u_{01}^5, u_{02}^5, \dots, u_{n,n+1}^5) \in \mathbb{R}^{n+1 \times n+1}$ are the dual variables associated to constraints (4.14). Given a dual solution $\bar{u} = (\bar{u}^1, \dots, \bar{u}^5)$ of the RMP, where we assume $\bar{u}_0^1 = \bar{u}_{n+1}^1 = 0$, the reduced cost of the column corresponding to a route (v_0, v_1, \dots, v_k) is given by:

$$\begin{aligned} \bar{z}_{SP} = & \sum_{j=0}^{k-1} \left(c_{v_j v_{j+1}} - (+1) \bar{u}_{v_j}^1 - (q_{v_{j+1}} + Q) \bar{u}_{v_j v_{j+1}}^4 - (s_i + t_{ij} + M_{ij}) \bar{u}_{v_j v_{j+1}}^5 \right) \\ & - \delta(v_0 \neq 0) \bar{u}_{v_0}^2 + \delta(v_k \neq n+1) \bar{u}_{v_k}^2 - \delta(v_0 = 0) \bar{u}^3. \end{aligned}$$

Regarding the reduced cost of a column in the set partitioning formulation, we can observe that more dual information is provided for the subproblem in a p -step formulation. Indeed, the shadow prices of resources are provided by the RMP and can be used to guide the decision at the subproblem level. In addition, this information can be used with no extra cost in the subproblem, as the duals are defined for pair of nodes and hence can be included as additional costs on the edges of the network. Therefore, even though p -step formulations have additional constraints in the MP with respect to SP formulations, we can ensure that they are robust [12], as the difficulty of solving the subproblem will be the same as in the SP formulation.

5 Advantages and disadvantages of the p -step formulations

We address now a few advantages of the p -step formulations with respect to the classical formulation for vehicle routing problems. As mentioned before, different types of capacity

constraints and valid inequalities can be incorporated to these formulations. They can be even stated in terms of partial paths instead of arcs, when it leads to stronger versions. Other advantage is that different requirements can be imposed directly to the master problem, which can be very convenient when dealing with rich vehicle routing problems and integrated problems, such as location routing and inventory routing problems [4, 8].

At the subproblem level, p -step formulations may lead to a better performance, as the label extension is limited by a new resource, the number of steps. Also, the subproblems can be solved in parallel, by splitting the label extension by starting node. Then, we are able to solve $n + 1$ subproblems in parallel, which is suitable for the current multi-core CPU machines.

As proposed in this paper, the size of the partial paths in the p -step formulation is limited by the number of traversed arcs. Since this can be seen as a resource, other types of resource may be used to limit a path: capacity, timing, etc. For example, we could generate partial paths in which the maximum load is a percentage of the capacity, or the total travel time is less than a percentage of the final time instant. This is somehow a generalization of the bidirectional label-extension [27], but with the joining of paths done at the master problem. This allows for any type of partitioning in the label-extension, instead of using two partitions only (e.g. several partitions of time).

Of course, p -step formulations have a few disadvantages as well. The first one is that the quality of the bound provided by the linear relaxation of a p -step formulation depends on p . A large p leads to a bound as good as that obtained from the SP formulation, while $p = 1$ leads to the weak linear relaxation of the VF formulation. A good strategy would be the use of a *turning point strategy*, in which the value of p is increased during the solution process. Hence, at the turning point, the p -step formulation is converted to a $(p + k)$ -step formulation, $k \geq 1$, by explicitly combining p -step paths to obtain $(p + k)$ -step paths.

Another disadvantage of the p -step formulation is related to the size of the master problem. Instead of the usual n constraints of the SP formulation, a p -step model of the VRPTW has $4n + 2n^2$ constraints, like in the VF formulation. Although the current linear programming solvers are very powerful nowadays, the solution time can be relatively large for n large. Nevertheless, interior point methods can help to overcome this weaknesses, specially if aided by active set strategies for identifying inactive constraints [14, 21, 15].

6 Conclusions

In this working paper, we have introduced a general class of formulations for vehicle routing problems, namely the p -step formulations. They offer several advantages over classical formulations and seem to be promising in practice. Theoretical results presented in this paper show that the classical formulations are special cases of the p -step formulations. Also, the proposed formulation can be put in a column generation scheme that allows more dual information to be sent to the subproblem.

This is still a ongoing research, in its very early stage. A computational implementation of a branch-price-and-cut method for the p -step formulation is in course and should be finished

soon. Computational results will be reported in a future version of this working paper.

Acknowledgments

The author is thankful to Claudio Contardo and Silvio Araujo for pointing out references [24] and [11] in private communication.

References

- [1] R. Baldacci, E. Bartolini, and A. Mingozzi. An exact algorithm for the pickup and delivery problem with time windows. *Operations research*, 59(2):414–426, 2011.
- [2] R. Baldacci, A. Mingozzi, and R. Roberti. Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. *European Journal of Operational Research*, 218(1):1–6, 2012.
- [3] J. E. Beasley and N. Christofides. An algorithm for the resource constrained shortest path problem. *Networks*, 19(4):379–394, 1989.
- [4] A. Ceselli, G. Righini, and M. Salani. A column generation algorithm for a rich vehicle-routing problem. *Transportation Science*, 43(1):56–69, 2009.
- [5] A. Chabrier. Vehicle routing problem with elementary shortest path based column generation. *Computers and Operations Research*, 33(10):2972 – 2990, 2006.
- [6] C. Contardo, G. Desaulniers, and F. Lessard. Reaching the elementary lower bound in the vehicle routing problem with time windows. *Networks*, 65(1):88–99, 2015.
- [7] G. Desaulniers, F. Lessard, and A. Hadjar. Tabu search, partial elementarity, and generalized k-path inequalities for the vehicle routing problem with time windows. *Transportation Science*, 42(3):387–404, 2008.
- [8] G. Desaulniers, J. G. Rakke, and L. C. Coelho. A branch-price-and-cut algorithm for the inventory-routing problem. *Transportation Science*, 2015.
- [9] M. Desrochers. An algorithm for the shortest path problem with resource constraints. Technical report, Technical Report G-88-27, GERAD, 1988.
- [10] D. Feillet, P. Dejax, M. Gendreau, and C. Gueguen. An exact algorithm for the elementary shortest path problem with resource constraints: application to some vehicle routing problems. *Networks*, 44:216–229, 2004.
- [11] I. Fragkos, Z. Degraeve, and B. De Reyck. A horizon decomposition approach for the capacitated lot-sizing problem with setup times. *INFORMS Journal on Computing*, 28(3):465–482, 2016.
- [12] R. Fukasawa, H. Longo, J. Lysgaard, M. P. de Aragão, M. Reis, E. Uchoa, and R. F. Werneck. Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Mathematical programming*, 106(3):491–511, 2006.

- [13] B. L. Golden, S. Raghavan, and E. A. Wasil. *The vehicle routing problem: latest advances and new challenges*, volume 43. Springer, 2008.
- [14] J. Gondzio, P. Gonzalez-Brevis, and P. Munari. New developments in the primal-dual column generation technique. *European Journal of Operational Research*, 224(1):41–51, 2013.
- [15] J. Gondzio, P. González-Brevis, and P. Munari. Large-scale optimization with the primal-dual column generation method. *Mathematical Programming Computation*, 8(1):47–82, 2016.
- [16] S. Irnich and G. Desaulniers. Shortest path problems with resource constraints. In G. Desaulniers, J. Desrosiers, and M. M. Solomon, editors, *Column Generation*, pages 33–65. Springer US, 2005.
- [17] S. Irnich, P. Toth, and D. Vigo. The family of vehicle routing problems. In P. Toth and D. Vigo, editors, *Vehicle routing: Problems, methods, and applications*, MOS/SIAM Ser Optim, pages 1–33. 2014.
- [18] G. Laporte, P. Toth, and D. Vigo. Vehicle routing: historical perspective and recent contributions. *EURO Journal on Transportation and Logistics*, 2(1-2):1–4, 2013.
- [19] M. E. Lübbecke and J. Desrosiers. Selected topics in column generation. *Operations Research*, 53(6):1007–1023, 2005.
- [20] R. Martinelli, D. Pecin, and M. Poggi. Efficient elementary and restricted non-elementary route pricing. *European Journal of Operational Research*, 239(1):102–111, 2014.
- [21] P. Munari and J. Gondzio. Using the primal-dual interior point algorithm within the branch-price-and-cut method. *Computers & Operations Research*, 40(8):2026 – 2036, 2013.
- [22] P. Munari and J. Gondzio. Column generation and branch-and-price with interior point methods. *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, 3(1), 2015.
- [23] D. Pecin, A. Pessoa, M. Poggi, and E. Uchoa. Improved branch-cut-and-price for capacitated vehicle routing. In *Integer programming and combinatorial optimization*, pages 393–403. Springer, 2014.
- [24] B. Petersen and M. K. Jepsen. Partial path column generation for the vehicle routing problem with time windows. In *International Network Optimization Conference (INOC), 2009,*, pages 1–6, 2009.
- [25] M. Poggi and E. Uchoa. New exact algorithms for the capacitated vehicle routing problem. In P. Toth and D. Vigo, editors, *Vehicle routing: Problems, methods, and applications*, MOS/SIAM Ser Optim, pages 59–86. 2014.

- [26] L. D. P. Pugliese and F. Guerriero. A survey of resource constrained shortest path problems: Exact solution approaches. *Networks*, 62(3):183–200, 2013.
- [27] G. Righini and M. Salani. New dynamic programming algorithms for the resource constrained elementary shortest path problem. *Networks*, 51(3):155–170, 2008.
- [28] F. Semet, P. Toth, and D. Vigo. Classical exact algorithms for the capacitated vehicle routing problem. In P. Toth and D. Vigo, editors, *Vehicle Routing: Problems, Methods, and Applications*, MOS/SIAM Ser Optim, pages 37–58. 2014.
- [29] M. M. Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2):254–265, 1987.
- [30] P. Toth and D. Vigo. *Vehicle Routing: Problems, Methods and Applications*. MOS-SIAM Series in Optimization, Second edition, 2014.